Efficient Make Before Break Capacity Defragmentation

Huy Duong, Brigitte Jaumard, David Coudert, Ron Armolavicius

Efficient Make Before Break Capacity Defragmentation

Huy Duong and Brigitte Jaumard
Computer Science and Software Engineering
Concordia University
Montreal, Canada
bjaumard@cse.concordia.ca

David Coudert
Université Côte d'Azur
Inria, CNRS, I3S
Sophia Antipolis, France
david.coudert@inria.fr

Ron Armolavicius
CIENA
Ottawa, Canada

***Abstract*—Optical multilayer optimization continuously reorganizes layer 0-1-2 network elements to handle both existing and dynamic traffic requirements in the most efficient manner. This delays the need to add new resources for new requests, saving CAPEX and leads to optical network defragmentation.**
The focus of this paper is on Layer 2, i.e., on capacity defragmentation at the OTN layer when routes (e.g., LSPs in MPLS networks) are making unnecessarily long detours to evade congestion. Reconfiguration into optimized routes can be achieved by re-defining the routes, one at a time, so that they use the vacant resources generated by the disappearance of services using part of a path that transits the congested section.
For the Quality of Service, it is desirable to operate under Make Before Break (MBB), with the minimum number of rerouting. The challenge is to identify the rerouting order, one connection at a time, while minimizing the bandwidth requirement.
We propose an exact and scalable optimization model for computing a minimum bandwidth rerouting scheme subject to MBB in the OTN layer of an optical network. Numerical results show that we can successfully apply it on networks with up to 30 nodes, a very significant improvement with the state of the art. We also provide some defragmentation analysis in terms of the bandwidth requirement vs. the number of reroutings.

Keywords*—Network reconfiguration, rerouting, defragmentation, make-before-break.

I. Introduction

Network reconfiguration is required in order to adapt to traffic changes, network failures, or new deployment of network resources. It occurs at the optical layer in order to make sure that the upper layer traffic, e.g., IP layer traffic, can be efficiently carried. In such a case, we deal with lightpath reconfigurations and the primary objective is to reduce disruptions to user traffic carried by existing lightpaths, measured by the number of disrupted lightpaths or the duration of lightpath disruptions [1]. Network reconfiguration may also appear in the OTN (Optical Transport Network) layer, in order to attain a better resource utilization [2]. In heavily loaded networks, dynamic connection addition and drop actions may result in a set of connections where some paths are not the shortest possible ones, leading to poor resource utilization compared to an optimal or at least optimized state. Thus, global connection re-optimization is proposed at certain time intervals (e.g., daily, weekly) to improve the network performance.

Researchers have investigated this connection re-optimization along two directions. In the first one, it consist in computing an optimized provisioning, and then find a sequence of connection rerouting in order to migrate from the current network provisioning to the optimized one with the minimum number of disruptions. In the second direction, the idea is to compute the best rerouting that allows moving away from the current provisioning with no disruption. While many studies have investigated the first direction, very few looked at the second one. In this paper, we propose to explore the second direction, and propose a scalable optimization model and algorithm in order to reach the best possible network provisioning subject to no disruptions, i.e., under the so-called Make Before Break (MBB) paradigm.

The paper is organized as follows. We briefly review the related papers to defragmentation in the OTN layer, in the next section (Section II), as well as the model of Klopfenstein [3], which is the only previously proposed optimization model for rerouting subject to MBB. We next describe in Section III our proposed decomposition model, called DEFRAG_RC, which requires in practice a much smaller number of variables and constraints than the model of Klopfenstein [3]. In Section IV, we explain how to solve efficiently the proposed DEFRAG_RC model with the DEFRAG_MBB algorithm, which contains a polynomial algorithm for the generation of the rerouting configurations. Numerical results are presented in Section V. Conclusions are drawn in the last section.

II. State of the Art

Note that we focus on Layer 2, while being aware that a lot of work has been recently made on Layer 0 in the context of flexible optical networks. But capacity defragmentation differs from spectrum defragmentation as there is no need to take care of continuity or contiguity constraints, and therefore we omit references related to Layer 0.

A. Literature Review

Several studies have been devoted to network reconfiguration with the minimum number of disruptions, following the strategy of migrating from a legacy ineffective routing to a given optimized one.
These studies usually have the constraint that a request can be rerouted at most once (i.e., from legacy to optimized route), together with the requirement of finding an optimized rerouting that is as close as possible to the legacy one. As a result, it usually prevents the existence of a strategy using only MBB

due to the presence of dependency cycles (e.g., request k_1 needs to be rerouted before k_2 because a link of the new route of k_2 belongs to the current route of k_1, and for similar reasons, k_2 needs to be rerouted before k_1). In order to find a rerouting strategies, authors have proposed to use the break-before-make (BBM) paradigm that allows for the temporary interruption of requests, and so breaking dependency cycles. For instance, Jose and Somani [4] propose heuristics for minimizing the total number of BBMs used in the rerouting strategy, and Coudert *et al.* [5], [6] and Solano and Pióro [7] provide scalable exact algorithms to minimize the concurrent number of BBMs. Tradeoffs between these two conflicting objectives are investigated by Cohen *et al.* [8] and Solano [9].
To further reduce the total or concurrent number of BBMs, Kadohata *et al.* [10] propose to use spare wavelengths to reroute a request to a temporary route rather that using a BBM. Said differently, they save one BBM by performing two MBBs. The network reconfiguration problem has also been investigated in logical layers, e.g., for MPLS networks [11], [12], [2], with the same constraints and objectives as above.
On the other hand, while many studies have investigated rerouting strategies both at the optical and the logical layer, very few studies have considered rerouting subject to the MBB paradigm (i.e., joint computation of optimized routing and rerouting strategy subject to MBB). Klopfenstein's study [3] is the only one proposing an optimization model, but unfortunately it is not scalable. We recall it in the next section as an introduction for our decomposition model in Section III.

B. Notations

We consider a network represented by a directed multi-graph $G = (V, L)$ where V is the set of nodes (indexed by v) and L is the set of links (indexed by ℓ). Different links may exist between two nodes in order to model different logical links, with e.g., different types of traffic. We denote $\omega^-(v)$ (resp. $\omega^+(v)$) the set of incident links incoming to (resp. outgoing from) node $v \in V$. Let C_ℓ denote the transport capacity of link ℓ. Let K be the set of connection requests (indexed by k). Connection request $k \in K$ is characterized by its source (s_k), destination (d_k), and bandwidth requirement (b_k).

We assume that the network undergoes a series of connection request re-optimization at different time stamps (rerouting events). Let T (indexed by t) be the set of those time stamps, with $t = 0$ being the initial one.

C. Klopfenstein's Model (2008)

The model of Klopfenstein [3] consists in finding the best possible rerouting, while guaranteeing it can be reached within a Make Before Break (MBB) policy. Indeed, Klopfenstein [3] proposed a very general network resource utilization function subject to a parameter α and that can be written as follows:

$$\text{OBJ}_\alpha = \frac{1}{1-\alpha} \sum_{\ell \in L} \left(C_\ell - \overbrace{\sum_{k \in K} b_k x_{k\ell}}^{\text{link load}} \right)^{1-\alpha}, \qquad (1)$$

There are two particular cases of interest, for which the objective (1) is a linear function:

- $\alpha = 0$, the objective consists then in maximizing the overall spare capacity:

$$\max \quad \left(\text{OBJ}_0 = \sum_{\ell \in L} C_\ell - \sum_{\ell \in L} b_k \left(\sum_{k \in K} x_{k\ell} \right) \right) \qquad (2)$$

- $\alpha = \infty$, the objective consists then in maximizing the smallest link spare capacity,

$$\max \left(\min_{\ell \in L} \left(C_\ell - \sum_{k \in K} b_k x_{k\ell} \right) \right). \qquad (3)$$

In the sequel, we will adopt OBJ_0. The set of variables is defined as follows:
- $x^t_{k\ell} = 1$ if demand k uses link ℓ in its routing at step t, 0 otherwise.
- $\pi^t_k = 1$ if demand k is rerouted at step t, 0 otherwise.

These objectives and variables are decided by a set of below constraints:

$$\sum_{\ell \in \omega^-(v)} x^t_{k\ell} - \sum_{\ell \in \omega^+(v)} x^t_{k\ell} = \begin{cases} -1 & \text{if } v = s_k \\ 1 & \text{if } v = d_k \\ 0 & \text{otherwise} \end{cases} \qquad k \in K, v \in V, t \in T \qquad (4)$$

$$\sum_{k \in K} x^t_{k\ell} b_k \leq C_\ell \qquad \ell \in L, t \in T \qquad (5)$$

$$\sum_{k \in K} \pi^t_k \leq 1 \qquad t \in T \qquad (6)$$

$$x^t_{k\ell} - x^{t-1}_{k\ell} \leq \pi^t_k \qquad k \in K, \ell \in L, t \in T \qquad (7)$$

$$x^t_{k\ell} \in \{0, 1\} \qquad k \in K, \ell \in L, t \in T \qquad (8)$$

$$\pi^t_k \in \{0, 1\} \qquad k \in K, t \in T. \qquad (9)$$

Constraints (4) are flow constraints and used in order to establish a route for each request, at each time period. Due to the subsequent constraints, the set of paths at time t will differ by at most one path from time $t-1$.
Constraints (5) enforce the transport capacity constraints. Constraints (6) impose that at most one request per step is rerouted. Constraints (7) are redundant if $\pi^t_k = 1$, i.e., if request k is rerouted at step t so that the routing variables $x^t_{k\ell}$ can be redefined, and otherwise impose that $x^{t-1}_{k\ell} = x^t_{k\ell}$, i.e., that routing is not modified. Indeed, note that if there is a rerouting, the two reroutings (legacy vs. optimized) must differ by at least one added and one dropped link. If $\pi^t_k = 0$, this is not allowed as $x^{t-1}_{k\ell} - x^t_{k\ell} \leq 0$, and consequently $x^{t-1}_{k\ell} = x^t_{k\ell}$. The last two sets of constraints define the domain of the variables. Note that if the rerouted path has a link in common with the original one, there is no need to double the capacity reservation corresponding to the considered demand [3].
Largest data instance on which experiments were conducted in [3]: a network with 10 nodes and about 40 links with capacity C_ℓ. Authors were not able to obtain optimal solutions for more than 5 reroutings within the time limit imposed (30 minutes).

III. A Decomposition Model: DEFRAG_RC

In this section, we propose a decomposition model, called DEFRAG_RC, based on a set of rerouting operations, where each rerouting operation proposes a potential MBB rerouting of a single connection request, i.e., a connection request for which there exists an alternate route with enough spare

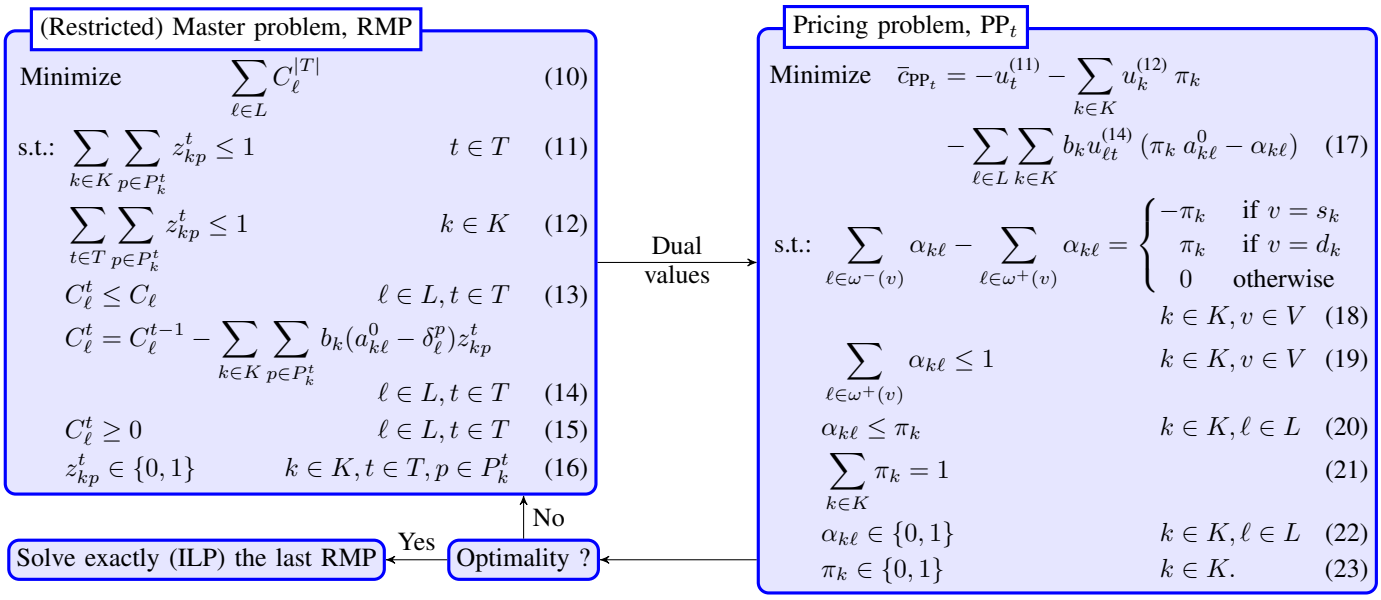


Fig. 1: Flow chart of decomposition model DEFRAG_RC

bandwidth for its routing. The model is parameterized by $|T|$, a bound on the number of rerouting operations. A solution of DEFRAG_RC is an ordered sequence of at most $|T|$ rerouting operations leading to the best provisioning reachable from the legacy provisioning with at most $|T|$ rerouting operations. Observe that less than $|T|$ rerouting operations might be sufficient to reach that optimized provisioning. The objective is to minimize the bandwidth requirements of the provisioning at time $|T|$. Observe that there is no guarantee to reach the best provisioning with a monotonous sequence, i.e., such that the overall bandwidth requirement decreases after each single rerouting operation (of a connection request). It may happen that the overall bandwidth requirement increases after a given rerouting operation in order to reach the minimum bandwidth provisioning at a later time, or at time $|T|$.

Let P be the overall set of potential rerouting operations, with $P = \bigcup_{k\in K}\bigcup_{t\in T} P_k^t$, where P_k^t is the set of routes of connection request $k\in K$ at time stamp $t\in T$.

The integer linear programming (ILP) formulation of DEFRAG_RC uses the following variables:

- $z_{kp}^t = 1$ if route $p\in P_k^t$ is selected at time stamp $t\in T$ for the rerouting of $k\in K$, 0 otherwise.
- C_ℓ^t = required bandwidth on link $\ell\in L$ at time stamp $t\in T$.

It also uses the following parameters:

- $a_{k\ell}^0 = 1$ if link $\ell\in L$ is used in the initial routing of connection request $k\in K$, 0 otherwise.
- $C_\ell^0 = \sum_{k\in K} b_k a_{k\ell}^0$ = initial bandwidth usage on link $\ell\in L$.
- $\delta_\ell^p = 1$ if path $p\in P$ uses link $\ell\in L$, 0 otherwise.

Objective: maximize the spare capacity

$$\max \sum_{\ell\in L}\left(C_\ell - C_\ell^{|T|}\right). \tag{24}$$

As $\sum_{\ell\in L} C_\ell$ is a constant value, the objective can be re-expressed as minimizing the bandwidth usage:

$$[\text{DEFRAG\_RC}] \quad \min \sum_{\ell\in L} C_\ell^{|T|} \quad \text{subject to} \quad (11)\text{-}(16).$$

Constraints (11) in Figure 1 prevent from selecting more than one rerouting operation at each time period. Note that $|T| \le |K|$ is an upper bound on the number of rerouting operation as we cannot predict a priori the number of required MBB reroutings. Constraints (12) ensure that a connection request is rerouted at most once. Constraints (13) make sure that transport capacities are never exceeded at any time stamp. Constraints (14) update the bandwidth usage on link ℓ at time stamp t, taking into account the unique connection request that has been rerouted at t. Constraints (15)-(16) define the domain of the variables.

IV. Solution Process: the DEFRAG_MBB Algorithm

The model DEFRAG_RC has an exponential number of variables, and therefore column generation is required in order to efficiently solve its linear relaxation.

This technique consists of decomposing the original problem into a Restricted Master Problem - RMP - (i.e., model (10)-(16) with a very restricted number of variables) and one or several pricing problems - PPs. In the particular case of model (10)-(16), we will show in next section that the pricing problem can be decomposed into $|K|\times|T|$ independent smaller pricing problems, each denoted by PP$_t^k$. The RMP and the PP(s) are solved alternately. Solving the RMP consists in selecting the best connection reroutings, while solving the PPs allows the generation of improving potential reroutings, i.e., such that, if added to the current RMP, improve the optimal value of its linear relaxation. The process continues until the optimality condition is satisfied, that is, all the so-called reduced costs defining the objective function of the pricing problems are positive. An ε-optimal solution is derived by solving exactly

the ILP model associated with the last RMP, with ε defined as follows:

$$\varepsilon = \left(\tilde{z}_{\text{ILP}} - z^{\star}_{\text{LP}}\right)/z^{\star}_{\text{LP}}, \tag{25}$$

where $z^{\star}_{\text{LP}}$ and $\tilde{z}_{\text{ILP}}$ denote the optimal LP value and the optimal ILP value of the last RMP, respectively.

Pricing Problem, PP$_t$

Let $u^{(11)}_t \leq 0$, $u^{(12)}_k \leq 0$ and $u^{(14)}_{\ell t} \gtreqless 0$ be the values of the dual variables associated with Constraints (11), (12) and (14), respectively. We use the following binary variables:

- $\pi_k = 1$ if $k \in K$ is selected for rerouting, 0 otherwise.
- $\alpha_{k\ell} = 1$ if $\pi_k = 1$ and the route of $k \in K$ uses link $\ell \in L$, 0 otherwise.

The goal of PP$_t$ is to select a unique request for potential rerouting, the one with a new route of minimum cost (Objective (17) in Figure 1). Constraints (18) take care of identifying the best possible route, using flow constraints, for the request that is rerouted, i.e., the unique request k such that $\pi_k = 1$. Constraints (19) make sure that we only output simple path, with no loops. Constraints (20) make sure that routing variables $\alpha_{k\ell}$ are null if request k is not selected for rerouting during time stamp t, i.e., the time period associated with the PP$_t$ pricing problem. Constraints (21) ensure that each PP$_t$ selects exactly one connection for potential rerouting at time stamp t. Constraints (22)-(23) define the domain of the variables.

Elementary Pricing Problem PP^k_t

Each PP$_t$ can be decomposed into $|K|$ elementary pricing problems PP^k_t, each examining the option of rerouting request $k \in K$ at time period t by setting $\pi_k = 1$ in PP$_t$. Then, the solution of PP$_t$ is given by

$$\overline{c}_{\text{PP}_t} = \min_{k \in K} \{\overline{c}^k_{\text{PP}_t} : k \text{ is rerouted at time period } t\}, \tag{26}$$

where $\overline{c}^k_{\text{PP}_t}$ denotes the reduced cost of PP^k_t:

$$\min\ \overline{c}^k_{\text{PP}_t} = 0 - u^{(11)}_t - u^{(12)}_k - \sum_{\ell \in L} b_k u^{(14)}_{\ell t}(a^0_{kl} - \alpha_{k\ell}) \tag{27}$$

subject to:

$$\sum_{\ell \in \omega^-(v)} \alpha_{k\ell} - \sum_{\ell \in \omega^+(v)} \alpha_{k\ell} = \begin{cases} -1 & \text{if } v = s_k \\ 1 & \text{if } v = d_k \\ 0 & \text{otherwise} \end{cases}, v \in V \tag{28}$$

$$\sum_{\ell \in \omega^+(v)} \alpha_{k\ell} \leq 1 \qquad v \in V \tag{29}$$

$$\alpha_{k\ell} \in \{0,1\} \qquad k \in K, \ell \in L. \tag{30}$$

Now, observe that PP^k_t is a weighted shortest *simple* path problem in a graph with possibly negative weight cycles. This problem is NP-hard by a reduction from the longest simple path problem (see [13, Section 24.1]), and so cannot be solved using only the Bellman-Ford-Moore (BFM) algorithm, even if it takes care of the negative weights. However, we can still use the BFM algorithm, but with some additional tool. As we cannot enforce the simple path condition, the BFM algorithm may fail to output a path with a negative reduced cost, due to the discovery of a negative cycle. In such a case, we then recourse to the solution of the ILP formulation of PP^k_t ((27)-(30)), which includes constraints to enforce the simple path condition. While we did encounter negative loops, it was quite rare and the solution of pricing problems with an ILP solver did not hamper much the overall computational times. It is worth noting that calls to the BFM algorithm can be grouped by sources. So $|V|$ calls to BFM suffice to solve PP$_t$.

Theorem 1: All pricing problems can be investigated with at most $O(|V|)$ runs of the BFM algorithm, leading to an $O\left(|L| \times (|K| + |V|^2) \times |T|\right)$ time complexity.

Proof: Note that the reduced cost in (27) can be rewritten:

$$\overline{c}^k_{\text{PP}_t} = \underbrace{-u^{(11)}_t - u^{(12)}_k - \sum_{\ell \in L} b_k u^{(14)}_{\ell t} a^0_{kl}}_{\text{constant}} + \sum_{\ell \in L} b_k u^{(14)}_{\ell t} \alpha_{k\ell}. \tag{31}$$

This entails that the solution of PP^k_t can be reduced to the solution of:

$$[\text{PP}^{\text{generic}}_{kt}] \qquad \min \sum_{\ell \in L} u^{(14)}_{\ell t} \varphi_\ell \quad \text{subject to:} \quad (28) - (30).$$

Observe that problem PP$^{\text{generic}}_{tk}$ is equivalent to a shortest simple path problem with negative weights, without any guarantee that it contains no negative cycles. Taking into account that *(i)* the coefficients of the objective function are independent of k, and *(ii)* the BFM algorithm can be easily modified in order to output a shortest path tree from a given source node, see, e.g., [13], (i.e., it computes all the (weighted) shortest paths from a given source node), we can then use $|V|$ calls of the BFM algorithm, one from each possible source node, for a given t. Then, for each connection k, we can compute $\overline{c}^k_{\text{PP}_t}$ using (31) and check whether the reduced cost is negative, and, if so, generate a new potential rerouting. For a given t, computing $\overline{c}^k_{\text{PP}_t}$ for all k can be done in $O(|K| \times |L|)$, once all $|V|$ BFM calls have been made, and each call to BFM requires time $O(|V| \times |L|)$, hence the overall complexity $O\left(|L| \times (|K| + |V|^2) \times |T|\right)$. ■

Solution Process

In order to be done efficiently, the solution of the $|K|$ Elementary Pricing Problem (PP^k_t) for a given t require their grouping as seen in the previous paragraph. In the context of a column generation model with a large set of different pricing problems, the efficiency of the solution depends on the best combination of linear program re-optimization (i.e., solution of the current Restricted Master Problem), and the solution of the whole set or a subset of pricing problems. We solve all the elementary PP^k_t associated with a given t, and add to the RMP the rerouting associated with the smallest reduced cost. We then perform a round robin on t.

V. Numerical Results

A. Data Sets

We consider a network with 32 nodes and 250 directed links, which corresponds approximately to a Ciena customer network. Existing network connections were used to construct a traffic matrix input to a simulation generating realistic random connection states. Connection requests had Poisson arrivals based on the traffic matrix and random durations drawn from a common exponential distribution. Each connection had a Weibull distributed bandwidth with a coefficient of variation

of 0.3. Connections were routed on the shortest path (in hop count) that had sufficient bandwidth. A load factor parameter was used to globally vary the connection arrival rates: the corresponding equilibrium connection states represent a range of congestion levels from light to heavy. For each load factor, we considered 10 defragmentation events. Defragmentation was performed with a period of 1 mean connection duration ensuring that sufficient connection requests and termination events occurred to produce comparably degraded pre-fragmentation connection states.
Characteristics of the data sets are described in Table I, where for each load factor, we provide the average number of granted requests right before each defragmentation.

TABLE I: Characteristics of the data sets

Load factor	Number of requests
0.5	777.2
0.6	894.6
0.7	951.4
0.8	971.1
0.9	993.3
1.0	1,015.4

B. Comparison with the Model of Klopfenstein [3]

We compared the performance of our model and algorithm with the model of Klopfenstein [3]. We use a dataset with a load factor of 0.5 as differences were already quite significant for a small load as indicated by the results reported in Table II. Therein, we report the results for each of the 10 defragmentation events. Columns entitled DEFRAG_RC correspond to the results obtained with the DEFRAG_RC model and DEFRAG_MBB algorithm.
We observe that although the accuracy of the model of Klopfenstein [3] is better (i.e., 0% for the data sets with a 0.5% load factor), the computing times are significantly larger, indeed about 100 times longer. Moreover, the computing time ratio is increasing with the load factor, that is why we only report the results for a 0.5 load factor. Although our proposed model does not always reach the 0% solution accuracy, the reached accuracy is very good as it is only 0.3% on average. The number of rerouting operations is larger with the model of Klopfenstein [3] as a consequence of a larger bandwidth saving due to a higher accuracy, but the difference is rather small however (ratio of 1.3 on average).

C. Defragmentation Efficiency

All statistics computed in this section corresponds to averages computed on all 10 defragmentations performed for each loading factor.
We first analyze the efficiency of the generation process for rerouting configurations with the results reported in Table III. We observe that, as the traffic load is increasing, then the number of initial reroutings of the initial set of configurations decreases in percentage. This can be explained by the fact that there is less spare capacity, and therefore, the number of rerouting sequences is more limited, and therefore the reroutings to be performed and their ordering is more difficult to identify in a simple greedy algorithm. In terms of the overall number of generated reroutings, note than the number of selected ones is at most $|T|$. We observe that for the 0.5 load, we reach the minimum bandwidth requirement with around 60 reroutings on average, and with less than 160 reroutings on average for the 0.6 load.
In Table IV, we report on the accuracy of the solutions: it is rather stable with $|T|$, i.e., between 2% and 3%, which is satisfactory taking into account the computational additional cost it would require for getting an optimal solution with a branch-and-price solution, instead of the current solution process. Computational times are also quite reasonable, although too long for a real-time defragmentation operation. However, we expect to reduce them significantly in the near future with the addition of heuristics to speed up the solution process.

D. Defragmentation Performance

We investigated the reduction of the overall bandwidth requirements after each defragmentation, and report in Figure 2 the reduction at each defragmentation event for the two extreme loading factors, i.e., 0.5 and 1.0. As already anticipated with the results of Table IV, increasing $|T|$ for the 0.5 load does not help to reduce further the minimum bandwidth requirement. For the 1.0 load, we get around an additional 5% bandwidth requirement reduction when increasing $|T|$ from 60 to 100, and then again from 100 to 150.

VI. Conclusion

We have proposed a new model for progressive defragmentation, i.e., computing a sequence of make before break reroutings leading to the minimum bandwidth requirements. It corresponds to a huge improvement with respect to the previous model proposed by Klopfenstein [3] as it allows to reach up to 150 reroutings in less than a few hours, while the model of [3] was only scalable for toy problems.
We plan to investigate further the proposed model in order that it can handle the case of more than one rerouting per (selected) connection while minimizing the overall number of reroutings.

Acknowledgments

B. Jaumard has been supported by a Concordia University Research Chair (Tier I). H. Duong was supported by a MITACS & Ciena Converge Fellowship. D. Coudert was supported by ANR program "Investments for the Future" under reference ANR-11-LABX-0031.

References

[1] H. Li and J. Wu, "Survey of WDM network reconfiguration: topology migrations and their impact on service disruptions," *Telecommunication Systems*, vol. 60, pp. 349–366, Nov. 2015.

[2] B. G. Józsa and M. Makai, "On the solution of reroute sequence planning problem in MPLS networks," *Computer Networks*, vol. 42(2), pp. 199–210, 2003.

[3] O. Klopfenstein, "Rerouting tunnels for MPLS network resource optimization," *European Journal of Operational Research*, vol. 188(1), pp. 293–312, 2008.

[4] N. Jose and A. K. Somani, "Connection rerouting/network reconfiguration," in *Proceedings of IEEE/VDE Workshop on Design of Reliable Communication Networks - DRCN*, 2003, pp. 23–30.

[5] D. Coudert, F. Huc, D. Mazauric, N. Nisse, and J.-S. Sereni, "Reconfiguration of the routing in WDM networks with two classes of services," in *Conference on Optical Network Design and Modeling - ONDM*, 2009, pp. 1–6.

TABLE II: Comparative results with the model of Klopfenstein [3]

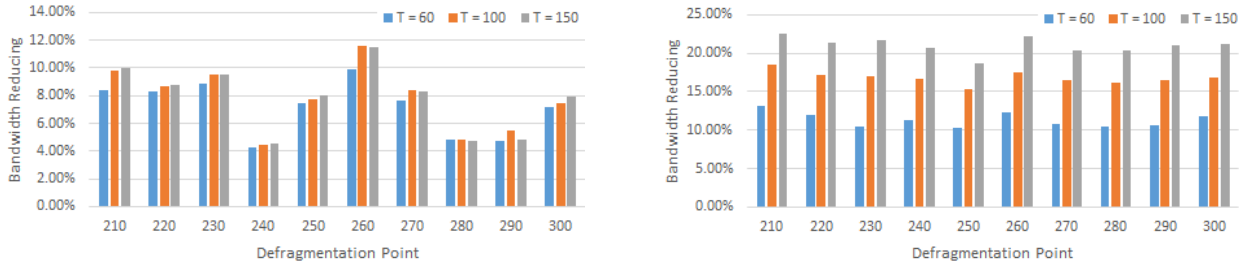
Defrag. event	Bandwidth saving DEFRAG_RC	Bandwidth saving Klopfenstein [3]	ε	# reroutings DEFRAG_RC	# reroutings Klopfenstein [3]	Computing times DEFRAG_RC	Computing times Klopfenstein [3]
1	3.2%	3.5%	0.3%	5	5	0.1	13.5
2	7.5%	8.2%	0.1%	13	15	0.2	11.8
3	9.9%	11.0%	0.0%	15	17	0.1	10.2
4	2.3%	2.7%	0.8%	6	15	0.2	29.0
5	9.6%	10.9%	0.7%	14	20	0.1	22.7
6	12.0%	13.7%	0.5%	16	18	0.3	17.1
7	6.6%	7.1%	0.2%	13	15	0.2	15.9
8	3.0%	3.1%	0.4%	5	7	0.1	17.4
9	6.7%	7.3%	0.3%	10	13	0.1	11.9
10	5.4%	5.7%	0.0%	7	7	0.1	2.9
Average	6.6%	7.3%	0.3%	10.40	13.20	0.15	15.2
Ratio	1.1			1.3		101.3	

TABLE III: Impact of the Initial Rerouting Configurations and Overall Number of Configurations

	# Initial configurations			# Initial configurations in optimal solution			Overall # of generated configurations		
Load factor	T = 60	T = 100	T = 150	T = 60	T = 100	T = 150	T = 60	T = 100	T = 150
0.5	47.6	50.7	50.7	25.7 (42.8%)	25.3	23.6 (25.7%)	1,308.2	2,338.1	3,465.1
0.6	60	100	132.6	26.0 (43.3%)	46.6	62.2 (26.0%)	1,254.0	2,169.6	3,669.5
0.7	60	100	150	23.4 (39.0%)	39.0	70.1 (23.4%)	1,187.4	1,956.3	2,782.4
0.8	60	100	150	21.9 (36.5%)	38.2	65.8 (21.9%)	1,245.5	2,074.3	2,798.7
0.9	60	100	150	19.1 (31.8%)	33.3	65.9 (19.1%)	1,173.7	1,906.9	2,804.2
1.0	60	100	150	18.8 (31.3%)	31.6	58.0 (18.8%)	1,274.7	1,996.3	2,806.2

TABLE IV: Number of Reroutings and Accuracy

	Number of required rerouting			Accuracy (ε)			Computation times (minutes)		
Load factor	T = 60	T = 100	T = 150	T = 60	T = 100	T = 150	T = 60	T = 100	T = 150
0.5	52.9	62.2	63.5	1.6	2.2	2.2	23.8	36.3	89.1
0.6	59.6	98.6	139.2	2.1	2.2	2.7	24.3	55.2	316.3
0.7	59.3	97.0	147.2	2.7	2.8	2.3	22.8	51.8	235.0
0.8	58.9	97.8	145.9	2.7	2.6	2.6	25.9	57.0	230.5
0.9	59.6	99.3	147.9	2.2	2.2	2.8	22.4	46.6	263.5
1.0	59.4	99.6	147.5	2.5	2.1	2.3	26.0	54.3	264.6



(a) Load factor = 0.5

(b) Load factor = 1.0

Fig. 2: Reduction (%) of bandwidth requirement

[6] D. Coudert, D. Mazauric, and N. Nisse, "Experimental evaluation of a branch-and-bound algorithm for computing pathwidth and directed pathwidth," *ACM Journal of Experimental Algorithmics*, vol. 21, no. 1.3, pp. 1–23, 2016.

[7] F. Solano and M. Pióro, "Lightpath reconfiguration in WDM networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 12, pp. 1010 – 1021, December 2010.

[8] N. Cohen, D. Coudert, D. Mazauric, N. Nepomuceno, and N. Nisse, "Tradeoffs in process strategy games with application in the WDM reconfiguration problem," *Theoretical Computer Science*, vol. 412, no. 35, pp. 4675–4687, 2011.

[9] F. Solano, "Analyzing two conflicting objectives of the WDM lightpath reconfiguration problem," in *IEEE Global Telecommunications Conference - GLOBECOM*, Nov. 2009, pp. 1–7.

[10] A. Kadohata, A. Hirano, F. Inuzuka, A. Watanabe, and O. Ishida, "Wavelength path reconfiguration design in transparent optical WDM networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 7, pp. 751 – 761, July 2013.

[11] S. Beker, D. Kofman, and N. Puech, "Off-line MPLS layout design and reconfiguration: Reducing complexity under dynamic traffic conditions," in *International Network Optimization Conference (INOC)*, October 2003, pp. 61–66.

[12] D. Coudert, D. Mazauric, and N. Nisse, "On rerouting connection requests in networks with shared bandwidth," *Electronic Notes in Discrete Mathematics*, vol. 32, pp. 109–116, 2009.

[13] T. Cormen, C. Leiserson, R. Rivest, and C. Stein, *Introduction to Algorithms*, 2nd ed. Cambridge: The MIT Press, 2001.